

# 控制与决策

Control and Decision

## 面向不确定性的多机器人路径鲁棒规划研究综述

张书凡, 毛剑琳, 张凯翔, 李睿祺, 李大炎, 王妮娅

引用本文:

张书凡, 毛剑琳, 张凯翔, 等. 面向不确定性的多机器人路径鲁棒规划研究综述[J]. *控制与决策*, 2024, 39(12): 3873–3888.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2023.1728>

## 您可能感兴趣的其他文章

### Articles you may be interested in

#### 移动机器人运动规划中的深度强化学习方法

Deep reinforcement learning for motion planning of mobile robots

控制与决策. 2021, 36(6): 1281–1292 <https://doi.org/10.13195/j.kzyjc.2020.0470>

#### 基于 $\text{pm}3\sigma$ 正态概率区间分族遗传蚁群算法的移动机器人路径规划

Path planning of mobile robot based on  $\text{pm}3\sigma$  normal probability interval population division using genetic ant–colony algorithm

控制与决策. 2021, 36(12): 2861–2870 <https://doi.org/10.13195/j.kzyjc.2020.0745>

#### 机器人信息增益RRT环境探索算法

Robot RRT based on information gain for environment exploration

控制与决策. 2021, 36(11): 2683–2689 <https://doi.org/10.13195/j.kzyjc.2020.1007>

#### 一种基于免疫机理的确定性移动机器人路径规划算法

A path planning algorithm of deterministic mobile robot based on immune mechanism

控制与决策. 2021, 36(10): 2418–2426 <https://doi.org/10.13195/j.kzyjc.2020.0059>

#### 基于免疫优化的平面Acrobot线性自抗扰鲁棒镇定

Robust stabilization of planar Acrobot using linear active disturbance rejection control with immune optimization

控制与决策. 2020, 35(12): 3053–3058 <https://doi.org/10.13195/j.kzyjc.2019.0289>

# 面向不确定性的多机器人路径鲁棒规划研究综述

张书凡<sup>1</sup>, 毛剑琳<sup>2†</sup>, 张凯翔<sup>1</sup>, 李睿祺<sup>1</sup>, 李大炎<sup>2</sup>, 王妮娅<sup>2</sup>

(1. 昆明理工大学 机电工程学院, 昆明 650500; 2. 昆明理工大学 信息工程与自动化学院, 昆明 650500)

**摘要:** 多机器人路径规划为共享工作环境中的多个机器人规划从起始位置到目标位置的无冲突路径集合, 是多智能体领域协同规划方向的重要课题. 然而, 时钟漂移、动态障碍物、机器故障等现实因素, 可能会导致多机器人系统的实际执行过程与规划方案存在时间上的偏差, 使机器人之间发生意外冲突. 面向不确定性因素展开多机器人路径鲁棒规划并在一定程度上保证规划方案的安全执行, 是当前路径规划领域的挑战之一. 对此, 首先整理不确定性下多机器人路径规划的问题描述; 然后分别从面向不确定性的鲁棒规划算法、在线执行策略以及评价指标等方面对目前研究进行综述, 讨论各类解决方法的思路和特点; 最后对多机器人路径鲁棒规划研究的关键技术进行展望.

**关键词:** 多机器人路径规划; 不确定性建模; 鲁棒规划; 冲突概率; 在线执行策略; 冲突和死锁避免

**中图分类号:** TP18 **文献标志码:** A

**DOI:** 10.13195/j.kzyjc.2023.1728

**引用格式:** 张书凡, 毛剑琳, 张凯翔, 等. 面向不确定性的多机器人路径鲁棒规划研究综述[J]. 控制与决策, 2024, 39(12): 3873-3888.

## Survey on robust multi-robot path planning under uncertainty

ZHANG Shu-fan<sup>1</sup>, MAO Jian-lin<sup>2†</sup>, ZHANG Kai-xiang<sup>1</sup>, LI Rui-qi<sup>1</sup>, LI Da-yan<sup>2</sup>, WANG Ni-ya<sup>2</sup>

(1. Faculty of Mechanical and Electrical Engineering, Kunming University of Science and Technology, Kunming 650500, China; 2. Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650500, China)

**Abstract:** Multi-robot path planning, aimed at finding conflict-free paths for multiple robots to navigate from their initial positions to their respective target positions within a shared working environment, is pivotal in multi-robot collaborative planning. However, practical factors such as clock drift, moving obstacles and machine faults may lead to temporal deviations between actual and planned execution processes of the multi-robot system, resulting in unexpected conflicts among robots. Robust multi-robot path planning under uncertainty, along with ensuring the safe execution of plans to a certain extent, is a research hotspot in the field of path planning. This article begins by summarizing the problem description of multi-robot path planning under uncertainty. It then reviews existing studies from perspectives of robust planning algorithms and online execution strategies as well as corresponding evaluation metrics. The logic and characteristics of various methods are discussed. Finally, the future research directions of the critical technologies in robust multi-robot path planning are put forward.

**Keywords:** multi-robot path planning; uncertainty modelling; robust planning; conflict probability; online execution policy; collisions and deadlocks avoiding

## 0 引言

随着机器人技术的不断发展, 各类多机器人系统在性能、灵活性和容错性方面得到显著增强<sup>[1]</sup>. 多机器人系统辅助人类执行模式化的繁重工作是未来智能化生产和服务的重要趋势, 目前在工业自动化<sup>[2]</sup>、仓储物流<sup>[3]</sup>、交通网络<sup>[4]</sup>、机场地面调度<sup>[5]</sup>

等领域已有广泛应用, 如图 1 所示. 多机器人路径规划<sup>[7]</sup>(multi-robot path planning, MRPP) 是多机器人系统中的关键挑战, 又称作多智能体路径找寻<sup>[8]</sup>(multi-agent path finding, MAPF), 其核心任务是为共享环境中的多台机器人规划从起点到目标点的无冲突路径集合. MRPP 的解决有助于合理分配和优化机器人的

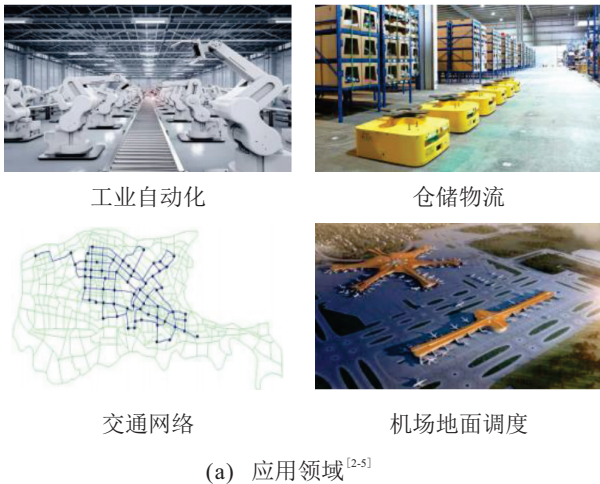
收稿日期: 2023-12-12; 录用日期: 2024-04-24.

基金项目: 国家自然科学基金项目(62263017).

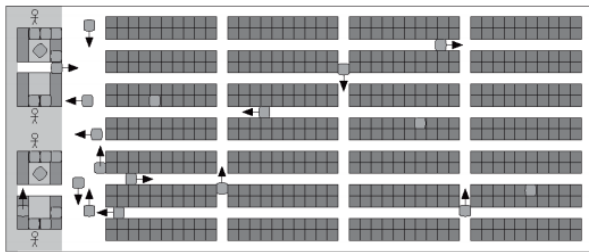
责任编委: 方勇纯.

<sup>†</sup>通讯作者. E-mail: jlmao@kust.edu.cn.

资源,促进机器人之间的协同合作,提高任务执行的效率.



(a) 应用领域<sup>[2-5]</sup>



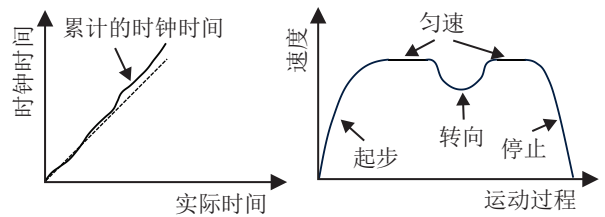
(b) 典型场景(以自动化仓储为例<sup>[6]</sup>)

图1 MRPP的应用领域及典型场景

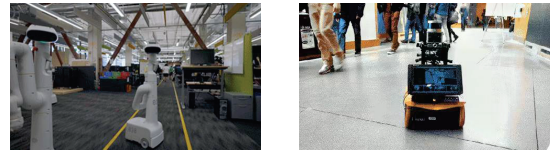
MRPP问题是一类NP-hard组合优化问题<sup>[9]</sup>,其状态空间随机器人数量呈指数级扩大,同时在有限空间内还需要解决机器人之间的冲突和死锁等问题.为了应对这一挑战,研究者们开发了分层协作A\*<sup>[10]</sup>、SIPP<sup>[11]</sup>、CBS<sup>[12]</sup>和优先级规划<sup>[13]</sup>等算法框架及其改进策略,如改进协作A\*<sup>[14-15]</sup>和改进CBS<sup>[16-17]</sup>等.这些算法可以为大规模机器人找到一组可行的路径规划,这种预先规划称为离线MRPP<sup>[18]</sup>.现有的离线MRPP算法可以保证机器人安全高效地抵达目标点,但前提是所有机器人都能在时间和空间上精确地遵循原定计划.

然而,在实际运行过程中机器人会受到诸多现实因素的影响,包括时钟漂移<sup>[19-20]</sup>、动力学约束<sup>[21]</sup>、执行误差<sup>[22]</sup>、电池损耗<sup>[23]</sup>、机器故障<sup>[24]</sup>等内部因素和复杂环境<sup>[25]</sup>、动态障碍物<sup>[26]</sup>、通信受限<sup>[27]</sup>等外部因素,以及机器人间的细微差别,如图2所示.

这些现实因素可能会导致机器人执行路径方案时产生时间上的不确定性,包括移动延迟<sup>[28]</sup>或者通过一段路径的行进时间出现波动<sup>[29]</sup>.这样的变化会使实际执行进程与既定计划产生偏离,使得路径有重叠的两个机器人可能发生冲突.同时,这种时间不确定性会不断累积,对路径方案的安全执行产生持续压力,最终可能导致系统崩溃.因此,在许多强调安全性



(a) 时钟漂移 (b) 运动学约束



(c) 复杂环境 (d) 动态障碍物

图2 机器人实际运行过程可能受到的影响

的实际场景中<sup>[30]</sup>,为保证多机器人路径规划在多种不确定性情况下的有效性,开展面向不确定性的多机器人路径鲁棒规划研究至关重要.

本文首先给出现有研究中面向不确定性的多机器人路径规划问题概述;其次,按照模型结构和优化目标对多机器人路径鲁棒规划算法进行分类总结,并阐述路径方案的3类在线执行策略,在此基础上整理鲁棒规划的评价指标;最后,对面向不确定性的多机器人路径鲁棒规划研究进行展望.

### 1 问题概述

多机器人在按照计划路线移动时,可能会遭遇不确定事件,如图3所示.根据计划路线的理论执行时间表,两个机器人在路线的重合点E3处不会产生冲突.然而在实际运行过程中,假设一个机器人需要在C3点之前等待以避让行人,而另一个机器人在E2点因临时故障停留了一段时间,则以上的等待和临时故障事件使两个机器人的移动过程产生延迟,进而可能在路径重合点E3发生冲突.

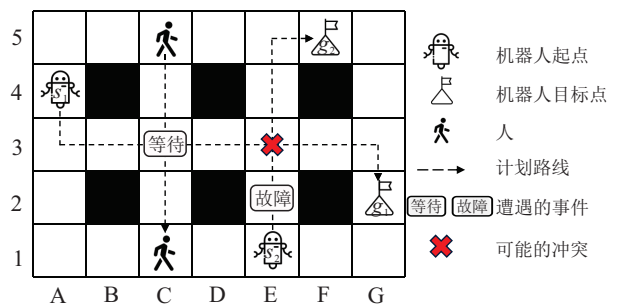


图3 多机器人移动中的潜在冲突示意图

多机器人的运行环境可以用无向图 $G = (V, E)$ 表示.其中: $V$ 是位置点的集合, $E$ 是位置点之间通路的集合.时间以离散时间步表示,记为 $t = 0, 1, \dots$ .假设有 $m$ 个带标签的机器人 $a_1, a_2, \dots, a_m$ ,每个机器人

$a_i$  都有给定的起始点  $s_i \in V$  和目标点  $g_i \in V$ . 机器人  $a_i$  的路径为  $\pi_i = [\pi_i(0), \pi_i(1), \dots, \pi_i(|\pi_i| - 1)]$ , 总的路径方案为  $\pi = [\pi_1, \pi_2, \dots, \pi_m]$ .

以时间代价总和或最大执行时间最小化为目标的优化模型如下:

$$\min \text{SOC}(\pi) = \min \sum_{i=1}^m (|\pi_i| - 1), \quad (1)$$

$$\min \text{Makespan}(\pi) = \min \max_{i=1}^m (|\pi_i| - 1). \quad (2)$$

$$\text{s.t. } \pi_i(t) \in V; \quad (3)$$

$$(\pi_i(t), \pi_i(t+1)) \in E \vee \pi_i(t+1) = \pi_i(t); \quad (4)$$

$$\pi_i(0) = s_i, \pi_i(|\pi_i| - 1) = g_i; \quad (5)$$

$$\pi_j(t) \neq \pi_i(t + \omega), \forall \omega \in \{0, 1, \dots, \Omega\}. \quad (6)$$

其中: 式(1)代表时间代价总和(sum of costs, SOC)的优化函数, 即最小化  $\pi$  中所有机器人从起始点到目标点所需的时间总和; 式(2)代表最大执行时间(Makespan)的优化函数, 即最小化  $\pi$  中单个机器人从起始点到目标点的最大时间代价; 式(3)代表机器人  $a_i$  在任意  $t$  时刻的位置  $\pi_i(t)$  必须在点集  $V$  中; 式(4)代表机器人在一个时间步内只能移动到相邻位置点或停留在原位置点; 式(5)代表路径必须自给定的起始点到给定的目标点; 式(6)是移动延迟下的冲突约束, 称为  $\Omega$ -延迟冲突约束, 该约束保证在延迟时间  $\omega \in \{0, 1, \dots, \Omega\}$  的情况下, 两个机器人间不会发生冲突. 若不满足式(6)约束, 则机器人间可能发生  $\Omega$ -延迟冲突. 延迟冲突约束是区别于经典 MRPP 问题的核心, 它能够保证不确定性延迟下的执行安全.

实际运行过程中, 机器人发生不确定性事件的种类、发生的位置、发生在哪个个体以及造成多长的延迟都是不确定的. 由于延迟具有分散性和累加性, 现有的建模方法主要将延迟分解为相邻节点间的、服从几何分布或 Gamma 分布的随机延迟时间.

离散时间设置下, 机器人在当前时间步计划移动但以  $p_d$  的概率移动失败, 于下一个时间步时仍处于当前节点, 这样的情况称为移动延迟<sup>[28,31-33]</sup>. 机器人从当前节点到下一个节点前发生移动延迟的次数便是单段延迟时间  $\Delta t$ , 且  $P(\Delta t = n) = p_d^n(1 - p_d)$  服从几何分布. 机器人从出发到第  $h$  个节点的累计延迟时间  $\Delta$  为

$$\Delta = \sum_{l=1}^{h-1} \Delta t_l. \quad (7)$$

如图4所示, 假设机器人  $a_1$  的路径  $\pi_1 = [s_1, A, B, C, g_1]$ , 机器人  $a_2$  的路径  $\pi_2 = [s_2, C, g_2]$ , 则由于  $\pi_2(1) = \pi_1(1 + 2) = C$  不满足式(6)约束, 该路径方案存在

2-延迟冲突. 即存在这样的可能: 若机器人  $a_2$  到达点  $g_2$  前累计延迟  $\Delta = 2$  个时间步, 而机器人  $a_1$  到达点  $g_1$  前未受延迟, 则两个机器人将在点 C 处产生冲突.

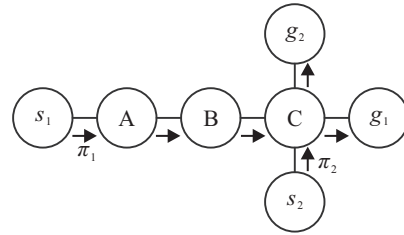


图4 几何分布延迟时间下的冲突

连续时间设置下, 需要对时间  $t$  进行扩充,  $t$  不只表示离散时间步, 还可以表示任意时间单位, 式(6)约束也将扩充为

$$\pi_j(t) \neq \pi_i(t + \omega), \forall \omega \in [0, \Omega]. \quad (8)$$

单段延迟时间  $\Delta t$  代表机器人在节点<sup>[34]</sup>或边上<sup>[25]</sup>遭遇的时间不确定性, 包括货物装卸、人机交互、避让动态障碍物花费的时间等.  $\Delta t \sim \text{Gamma}(\alpha, \beta)$ . 其中:  $\alpha$  为形状参数, 代表不同点或边上的不确定性程度;  $\beta$  为比率参数, 一般设为常数. 由于 Gamma 分布的可加性, 机器人从出发到第  $h$  个节点的累计延迟时间也服从 Gamma 分布, 即

$$\Delta \sim \text{Gamma}\left(\sum_{l=1}^{h-1} \alpha_l, \beta\right). \quad (9)$$

假设  $\pi_j(t_j) = \pi_i(t_i) = X$  且  $t_i > t_j$ , 机器人  $a_i$  和  $a_j$  到达点 X 前的累计延迟时间分别是  $\Delta_i$  和  $\Delta_j$ , 且离开点 X 时将继续延迟  $\Delta t_i$  和  $\Delta t_j$ . 定义事件 A:  $t_i + \Delta_i + \Delta t_i > t_j + \Delta_j$ ; B:  $t_j + \Delta_j + \Delta t_j > t_i + \Delta_i$ . 若 A 和 B 同时发生, 则机器人  $a_i$  和  $a_j$  对点 X 的占用时间有重叠. 令  $\Omega = \max(\Delta_i + \Delta t_i, \Delta_j + \Delta t_j)$ , 这种重叠不满足式(8)约束  $\pi_j(t) \neq \pi_i(t + \omega) = X$ , 其中  $\omega = t_i - t_j, \omega \in [0, \Omega]$ , 即存在  $\Omega$ -延迟冲突. 如图5所示, 分布  $D_i$  和  $D_j$  分别代表机器人  $a_i$  和  $a_j$  对点 X 占用时间的概率密度(服从 Gamma 分布), 可见由于延迟, 两个机器人对点 X 的占用在其原计划时间  $t_i$  及  $t_j$  的右侧都有分布, 因此在阴影区域中存在冲突可能.

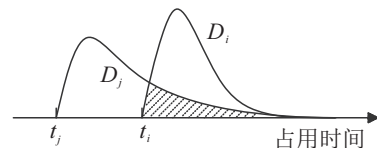


图5 Gamma分布延迟时间下的冲突

由不确定性因素引发的意外冲突问题, 目前主要在规划阶段和执行阶段进行解决, 如图6所示. 在规划阶段, 使用鲁棒规划算法生成对不确定性因素有一定容忍能力的路径方案, 使得系统在有限干扰下仍能

正常运行.其本质是通过牺牲一部分路径代价,松弛路径方案对理想化执行的要求,换取对随机干扰的容忍能力.但是其能应对的干扰通常有一个范围时,当超出这个范围时,鲁棒性随之失效.而在执行阶段,该问题可以得到一定的解决.在线执行策略可以应对较大或难以预测的干扰,通过实时协调,最大程度上保证系统的安全性.但是在线执行策略需要在协调代价和实时计算能力间取得平衡,简单的协调方式可能会造成较大的性能浪费,甚至使系统停摆.此外,两者结合使用时,鲁棒规划可以有效减轻在线执行的负担.

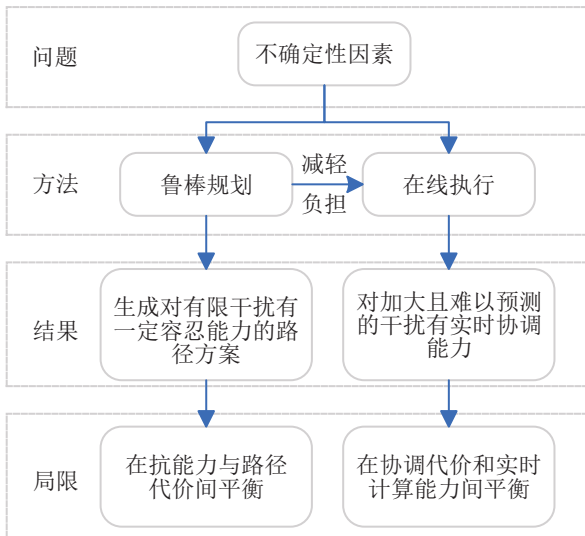


图6 多机器人路径鲁棒规划研究概况

## 2 面向不确定性的多机器人路径鲁棒规划算法

鲁棒规划算法基于不确定性的建模,在规划阶段便将执行时可能会遇到的不确定性考虑在内,并针对特定的优化目标规划出对不确定性有一定承受能力的方案<sup>[35]</sup>.这样的方案在执行过程中即使受到了一些不确定性事件的干扰,仍能保障机器人之间不发生冲突.针对各类不确定性场景和优化目标,学者们提出了多种面向不确定性的多机器人路径鲁棒规划算法,如表1所示.

### 2.1 基于时间膨胀模型的算法

时间膨胀模型通过增广机器人占用位置点或线路的时间来吸收可能遇到的延迟.膨胀后的时间通常为一个确定值,以简化参数设置,减轻计算负担,并且能得到确定的安全性(在2.2节中概率模型得到的是概率安全性).

#### 2.1.1 机器人占用时间的膨胀

$k$ -鲁棒路径规划<sup>[32]</sup>将机器人占用一个位置的时间由一个时间步膨胀为 $k$ 个时间步,通过禁止两个机

器人在相邻 $k$ 个时间步之内进入同一位置,保证每个机器人在执行过程中至多延迟 $k$ 个时间步而不发生冲突.如图7所示,若按照原始路径 $\pi_1 = [B, C]$ 及 $\pi_2 = [A, B]$ ,则机器人 $a_2$ 将在 $t = 1$ 时到达终点B.而 $k$ -鲁棒路径规划( $k = 1$ )将机器人 $a_1$ 在点B的占用时间多膨胀了1个时间步,不允许 $a_2$ 在 $t = 1$ 时进入点B,于是 $a_2$ 在点A停留直到 $t = 2$ 时才进入点B,避免了因 $a_1$ 可能在点B延迟一个时间步而产生的冲突.

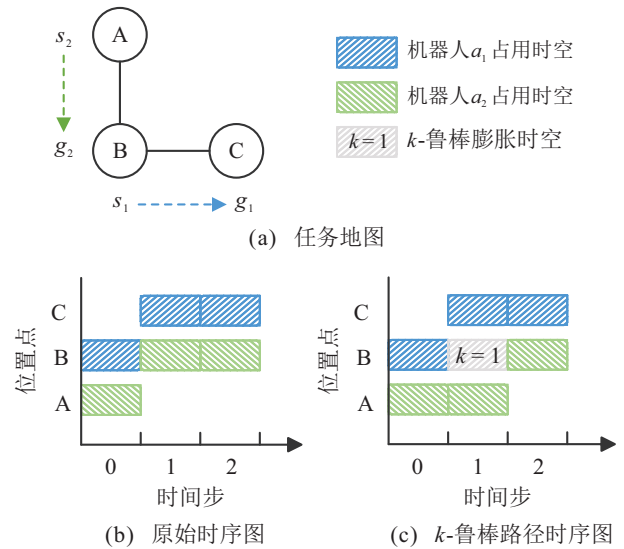


图7  $k$ -鲁棒路径规划图示

$kR$ -CBS算法用于求解最优的 $k$ -鲁棒路径方案,其与传统CBS的区别在于约束树的扩展. $kR$ -CBS通过引入 $k$ -延迟冲突,并设置相应的约束条件对约束树进行扩展.但传统CBS的约束只能限制机器人在某一个时间步的行为,所以常出现需要多个约束才能解决一个 $k$ -延迟冲突的情况.因此为了缩减约束扩展树的大小, $kR$ -CBS的变体算法Improved  $kR$ -CBS<sup>[32]</sup>引入了范围约束的概念,限制机器人在一整段时间范围内进入特定位置,一次性解决 $k$ -延迟冲突.范围约束设置得越大,约束树的扩展节点越少.但是过大的约束范围可能会导致算法失去完整性和最优性.因此在保证完整性和最优性的前提下寻找合适的约束范围,最大程度地减少约束树节点的扩展数量,是该算法的关键.在 $k$ -鲁棒路径规划中,可能会出现一些特殊冲突类型,即在一块局部区域内重复进入相似冲突.针对这一问题,文献[36]提出了 $k$ -延迟矩形冲突、 $k$ -延迟窄道冲突和 $k$ -延迟目标冲突的概念,并设计了相应的对称性冲突消解策略.这些策略的引入进一步减少了冗余的约束树分支,提高了 $kR$ -CBS算法的效率和可行性.另外, $kR$ -SIPP<sup>[37]</sup>是一种基于安全区间<sup>[11]</sup>(safe interval path planning, SIPP)的 $k$ -鲁棒路径规划算法,能够得到次优的 $k$ -鲁棒路径方案.它通过

表 1 面向不确定性的多机器人路径鲁棒规划算法分类

模型	优化目标	算法名称	特点	优点	缺点
时间膨胀模型	$k$ 个延迟内最优无冲突路径	$kR$ -CBS <sup>[32,36]</sup> , $kR$ -SIPP <sup>[37-38]</sup> , $k$ -Token Passing <sup>[39]</sup>	单个机器人的累计延迟次数较低	个体累计延迟在 $k$ 以内时, 直观有效地保证安全性	难以应对较高的个体累计延迟
	行进时间波动范围内最优无冲突路径	$A^*$ +OD <sub>TU</sub> <sup>[29]</sup> , CBS <sub>TU</sub> <sup>[29]</sup>	通过各条边的时间在范围内波动	在行进时间波动范围内保证安全性	保守情况下路径的代价总和较高
概率模型	单个机器人的最大冲突概率	UM* <sup>[33]</sup>	能够估计机器人处于各位置点的概率	路径方案易于在执行过程中使用在线修复或重规划来保持安全性	计算复杂度较高, 需要维护和搜索大量的信念状态
	全局执行成功率	$pR$ -CBS <sup>[31]</sup> , $p$ -Token, Passing <sup>[39]</sup>	需要保证全局执行成功率的场景	在个别机器人累计延迟次数较大的情况下仍能保证较高的执行成功率	计算负担大, 延迟较少时仍可能冲突
	两个机器人之间的冲突概率	STT-CBS <sup>[34]</sup> , greedy STT-CBS <sup>[25]</sup>	机器人在节点或通路上存在可预估的随机延迟	为随机行进时间下的路径规划提供了可量化的鲁棒性	可能会使个别机器人的冲突概率较高
	期望实际完工时间	AME <sup>[28]</sup>	优化因在线执行策略而延长的实际完工时间	执行过程中的期望完工时间较小	必须与在线执行策略组合使用
	满足特定运行规则	基于行进时间概率传播模型的优先级规划算法 <sup>[40]</sup>	需要优化时间利用率的实际场景	将行进时间在不同优先级之间传播, 算法执行速度快	应对强安全约束且动态变化的环境时, 缺乏安全性保证
时间无关模型	保证系统安全运行并避免死锁	时间无关离线路径规划算法 <sup>[19]</sup>	多个机器人可以异步执行, 且执行过程不需中心控制器	不需保持动作同步或时序关系	可解性较差
过渡系统模型	部分可观察下的行为决策	Dec-POMDP <sup>[41-43]</sup>	可以对不确定性环境下的决策过程建模	能够在部分可观察下形成状态估计和行为决策	状态空间复杂度高, 状态估计难度大
	满足安全规范和团队行为奖励最大化	基于广义随机 Petri 网的马尔可夫决策过程 <sup>[44]</sup>	匿名多机器人系统, 带有安全规范及团队行为奖励	能够根据团队整体行为要求, 推导出单个机器人的行为策略	只能处理线性约束和累积奖励, 而非更一般的团队行为规范
	减轻机器人交互对执行时间的影响	基于拥塞概率模型的单机器人马尔可夫决策过程 <sup>[45]</sup>	多个机器人可以同时经过一条边	优化拥塞路段选择	机器人无法正常通过拥塞路段时, 方案执行会受到很大影响
其他模型	根据实际执行情况切换备用路径	备用规划 <sup>[46]</sup> , 容错规划 <sup>[24]</sup> , 方案树 <sup>[47]</sup>	机器人只能观察到临近的冲突, 且不具备重规划能力	不需要在执行过程中进行重规划, 能快速切换备用路径且保证路径质量	难以保证执行过程中的可达性或需要庞大的方案树
	工作空间分解, 形成分块小规模问题	碎片系统 <sup>[48]</sup>	大规模空间中协调大量机器人, 能够应对移动失败或目标变化	降低问题求解复杂度, 可在碎片内部根据实际执行状态更新局部路径	在一些特殊的工作空间中不具备完备性

引入  $k$ -鲁棒因子扩展了 SIPP 中的障碍区间, 实现了 SIPP 的  $k$ -鲁棒化.  $kR$ -SIPP 的规划速度较  $kR$ -CBS 有了较大提升, 在此基础上,  $DkR$ -SIPP<sup>[38]</sup> 引入动态优先级策略来解决  $kR$ -SIPP 中存在的起步困境问题, 进一步提高了规划成功率.

基于  $k$ -鲁棒思想的  $k$ -Token Passing ( $k$ -TP) 算法<sup>[39]</sup> 用于求解考虑延迟的多机器人取送货物问题 (multi-agent pickup and delivery with delays, MAPD-d).  $k$ -TP 算法在机器人之间传递令牌以分配路径, 并为每个机器人保留  $k$  个时间槽作为缓冲区.  $kR$ -CBS 算法还被应用于求解多列车路径规划问题<sup>[49-50]</sup>, 其中每部列车都占据一系列位置, 在形式上与时间膨胀机器人类似.

$k$ -鲁棒路径规划适用于各个机器人受到的累计延迟较少且较平均的情况. 这种方法可以提供直观

的鲁棒性保证, 使机器人能够在一定延迟范围内安全地执行任务. 然而, 当累积延迟较多时,  $k$  值也需要相应地增大, 这将导致路径方案中冗余的等待时间增多, 从而使路径的总代价增大, 并且较大的  $k$  值也会使规划更加困难<sup>[31]</sup>.

### 2.1.2 单段线路行进时间的膨胀

将机器人通过一条边的行进时间由单个时间步膨胀为一个有上下界的时间步区间<sup>[29]</sup>, 可以简单表示离散行进时间在某个范围内的波动. 将路径上各段行进时间的上下界分别累加, 可以计算出机器人到达其路径各点的潜在时间范围. 如图 8 所示, 机器人从  $s$  点到  $g$  点路径中各段行进时间分别为  $[1, 3]$ 、 $[1, 1]$ 、 $[1, 2]$ , 则机器人到达各点的时间范围分别为  $t_{v_1} = [1, 3]$ 、 $t_{v_2} = [2, 4]$ 、 $t_g = [3, 6]$ . 然后通过计算路径各点在时间范围上的重叠, 即可检测出多机

机器人之间的潜在冲突.为了解决这些潜在冲突,文献[29]提出了A\*+OD<sub>TU</sub>和CBS<sub>TU</sub>两种算法寻找无潜在冲突的最优路径,并使用在线重规划缩短多余的等待时间.单段行进时间膨胀模型可以灵活反应机器人在不同位置之间行进时间的不确定性,并且设置简单,易于计算.但是行进时间上下界的设置需要合理选择,选择不当可能会降低算法的可靠性.

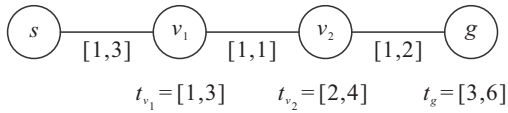


图8 单段行进时间膨胀图示<sup>[29]</sup>

## 2.2 基于概率模型的算法

机器人受到不确定因素的影响,其行进时间及所处位置从确定的变为概率分布的,因此使用概率模型求解问题是一种自然的选择.概率模型基于时间或位置上的概率分布得到冲突概率<sup>[31,33]</sup>及最大完工时间<sup>[28]</sup>等指标的估计,易于评估路径规划的可靠性及优化目标函数.

降低冲突概率是面向不确定性的MRPP问题的核心,也是概率模型的关键.基于信念空间的UM\*算法<sup>[33]</sup>将机器人的位置用路径上各点的概率分布进行描述,如图9所示.UM\*根据位置概率计算每个机器人遭遇冲突的概率,并求解个体冲突概率最小的路径.UM\*算法首先为多个机器人分别规划路径,然后只为存在冲突概率的机器人进行协调搜索,从而降低协调搜索的复杂度,是一种低耦合的规划算法.其变体PUM\*算法<sup>[33]</sup>引入了随机重启策略避免陷入局部最优.UM\*算法找到的方案易于在执行过程中使用在线修复或重规划来保持系统安全性,但是UM\*算法计算复杂度较高,在低耦合设置下仍需要维护和搜索大量的信念状态.

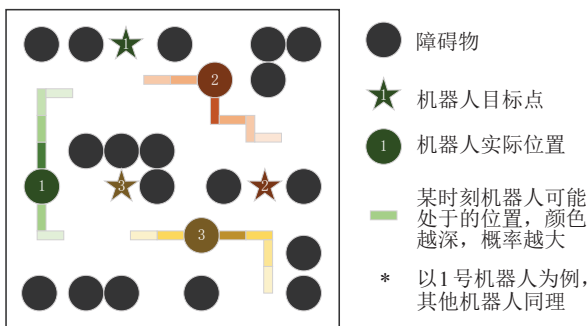


图9 机器人的信念状态<sup>[33]</sup>

先规划再优化是CBS算法的核心思想,传统CBS中优化对象是确定的冲突,通过添加约束逐步消除方案中的冲突.而在不确定性问题中,CBS变体<sup>[25,31,34]</sup>的优化对象是冲突概率,通过添加约束逐步

降低冲突概率. $p$ -鲁棒路径规划<sup>[31]</sup>的目的是寻找执行过程中完全不发生冲突的概率 $p_0$ (耦合的全局执行成功率)大于给定的 $p$ 值且代价最小的路径方案.确定性检验法和蒙特卡洛检验法可以基于单步延迟概率 $p_d$ 计算路径方案的执行成功概率 $p_0$ ,其中蒙特卡洛法更加简单高效. $p$ -鲁棒路径规划使用基于CBS的 $pR$ -CBS算法进行搜索,为了保证最优性, $pR$ -CBS的约束树扩展时有3个分支,这导致了较慢的扩展速度.因此, $pR$ -CBS的变体Greedy  $pR$ -CBS算法<sup>[31]</sup>选取 $p_0$ 最大的节点进行扩展,并且将约束分支削减到两个,以失去最优性为代价提高了搜索速度.实际应用中,结合路径执行失败带来的损失和执行成功带来的收益,可以通过损失-收益比得到所需 $p$ 值<sup>[31]</sup>.基于 $p$ -鲁棒路径规划的 $p$ -Token Passing ( $p$ -TP)算法<sup>[39]</sup>应用于求解考虑延迟的多机器人取送货问题(MAPD-d),类似地,基于CBS的STT-CBS算法<sup>[34]</sup>用于寻找考虑随机行进时间的MRPP问题的最优路径,同时保证每对机器人之间的冲突概率不超过指定阈值.该方法能够提供可量化的鲁棒性,但可能会使个别机器人的冲突概率较高,适用于机器人在节点或通路上存在可预估随机延迟的场景.STT-CBS算法扩展约束树节点时,分别延迟两个冲突机器人到达冲突位置的时间,使得该对机器人间的冲突概率小于阈值.STT-CBS具有与CBS算法相同的复杂度,但是计算冲突概率的过程会增加计算开销<sup>[34]</sup>.进一步,Greedy STT-CBS算法<sup>[25]</sup>在STT-CBS的基础上提高了约束树扩展速度.Greedy STT-CBS算法估算路径方案中任意两个动作之间的冲突概率,比较出冲突概率最大的动作对,并以它的冲突概率作为该方案的冲突概率,然后选择冲突概率最小的方案进行扩展.这种节点扩展的启发式函数与 $pR$ -Greedy CBS<sup>[31]</sup>类似.Greedy STT-CBS的搜索过程有时间限制,到达规定时间时算法返回具有最小冲突概率的路径方案.

在概率模型中,期望的实际完工时间可以得到优化.Approximate minimization in expectation (AME)算法<sup>[28]</sup>基于延迟概率和动作依赖估算实际完工时间的期望,并以之作为启发式函数引导基于CBS框架下的搜索过程,旨在减少因潜在的在线修复带来的等待时间,使期望的实际完工时间最小.需要注意的是,这种方法必须与在线执行策略配合使用,而无法单独为机器人的执行安全提供保障.此外,在不确定性下机场地面运行的路径规划过程中<sup>[40]</sup>,高优先级飞机的行进时间可以经概率传播马尔科夫链向低优先级飞机的传递,然后基于优先级顺序和传递的行进时间

为飞机规划路径.该方法根据机场地面的实际运行状况进行实时规划,可以提高安全性并减少等待时间.但是为应对强安全约束且动态变化的机场环境,同样需要与在线执行策略相结合.

### 2.3 基于时间无关模型的算法

传统MRPP问题大多基于路径方案的执行时间假设,需要机器人按照计划时间到达指定地点才能确保安全,这种方式对时间不确定性比较敏感<sup>[19]</sup>.因此文献[19]提出了与时间无关的离线多机器人路径规划问题(offline time-independent multiagent path planning, OTIMAPP),旨在消除方案执行对时间的依赖性.OTIMAPP对机器人到达某个位置或开始某个动作的时间不作要求,机器人只要沿着规划好的路径前进并在执行中保证位置互斥,最终都能到达目的地.也就是说OTIMAPP方案的执行允许任何有限延迟,允许机器人自然地以任意局部顺序通过路径重合位置,这与第3节中的修复策略不同.但是在执行过程中,由于机器人局部顺序的随机性,可能会导致死锁的发生.死锁就是一连串机器人形成了首尾循环阻碍,陷入死锁的机器人无法继续执行其计划.为此,OTIMAPP引入潜在死锁约束,如下所示:

$$\begin{aligned} & \forall (\{i, j, k, \dots, l\}, \{t_i, t_j, t_k, \dots, t_l\}), \\ & \neg(\pi_i(t_i + 1) = \pi_j(t_j) \wedge \pi_j(t_j + 1) = \\ & \pi_k(t_k) \wedge \dots \wedge \pi_l(t_l + 1) = \pi_i(t_i)). \end{aligned} \quad (10)$$

式(10)规定了OTIMAPP规划出的路径方案中,任意机器人子集和其时序的组合都不会导致循环阻碍.若不满足式(10),则存在特定的延迟情况会使 $\{i, j, k, \dots, l\}$ 号机器人陷入死锁状态.文献[19]提出了基于潜在死锁约束的两种算法求解OTIMAPP问题,包括优先级规划(prioritized planning, PP)和基于死锁的搜索算法(deadlock-based search, DBS).OTIMAPP方案的执行过程不需要同步,也可以不依赖中心控制器,仅通过本地交互来保证位置互斥即可.但是OTIMAPP的可解性较差,相同场景下传统MRPP可能有解但OTIMAPP无解,并且OTIMAPP中的死锁检测程序会带来较大的计算负担.

### 2.4 基于过渡系统模型的算法

过渡系统可以对多种不确定性建模,包括动作结果的不确定性<sup>[51]</sup>、部分可观察性<sup>[41-42]</sup>、时间不确定性<sup>[52]</sup>和机器人交互的影响<sup>[45]</sup>.过渡系统的基本要素是状态,以及状态之间的过渡动力学<sup>[53]</sup>.基于马尔可夫决策过程(Markov decision processes, MDP)的过渡系统可以捕捉机器人的异步执行和不确定的动作执

行时间<sup>[53-54]</sup>.分布式部分可观察MDP(decentralized partially observable MDP, Dec-POMDP)依据每个机器人的局部观测更新当前联合状态的概率分布<sup>[41-42]</sup>,但是其联合状态空间存在指数爆炸问题.为了减小状态空间,Dec-POMDP可以解耦成每个机器人的局部POMDP,并为每一个局部POMDP计算状态的概率分布,以构建单机器人MDP<sup>[43]</sup>.

广义随机Petri网(generalised stochastic petri nets, GSPNs)可以表示同类(匿名)机器人团队的状态、动作和不确定的动作时间<sup>[44]</sup>.通过设定安全约束和奖励并建立嵌入式MDP,可以求解机器人团队在GSPNs中的行为选择,以最大化团队的期望累积奖励,同时尽可能长时间地维持安全约束.然而,该方法同样受到状态空间指数增长的限制,对于较大的机器人团队或较复杂的环境可能难以求解,且只能处理线性约束和累积奖励,而非更一般的团队行为规范.目前GSPNs主要用于对足球机器人、自主搬运器和监视机器人团队的建模<sup>[53]</sup>.

特定场景中多个机器人可以同时存在于一条路径上,通过自身的运动规划能力避免碰撞.但是多个机器人在小范围内的交互会带来不确定的时间代价<sup>[45]</sup>.文献[45]使用相位类型分布(phase-type distributions, PTD)表示机器人通过一条边的行进时间,这是一种能够捕捉任意精度水平的非负指数序列<sup>[55]</sup>.它受到拥塞度,即当前边上同时存在的其他机器人数量的影响.规划过程中,为每个机器人按照优先级顺序依次构建单机器人MDP,该过程的转移函数是基于行进时间分布的.在完成一个机器人的规划后,更新地图上的拥塞度和行进时间分布.这种方法对机器人因运动协调带来的时间不确定性进行建模,使得在路径规划问题中可以容纳适度的拥塞,并综合考虑路径代价以做出决策,最终减少整体的任务执行时间.但是当机器人在拥塞路段上的协调不如预期时,方案执行会受到很大影响.

### 2.5 基于其他模型的算法

为应对执行过程中的未知情况,提前规划备用路径也是一种常见的方法<sup>[46-47,24]</sup>.文献[46]为每个机器人构建一条主路径和若干条备用路径,当机器人在执行过程中遇到延迟后,可以根据延迟情况选择切换到一条备用路径,避免与其他机器人发生碰撞.该方法不需要在执行过程中使用重规划,减少了计算开销,但是无法保证备用路径之间是无冲突的.类似地,文献[24]为每个机器人准备一组备用路径和切换规则,保障机器人当检测到周围存在冲突的机器人时可以

切换接下来的路径.该方法的主要思想是让每个机器人维护一个动态的约束图,表示它与其他机器人的依赖关系.当一个机器人崩溃时它会通知相邻的机器人,然后这些机器人会修复自己的约束图,并根据新的约束图选择备用路径.在不影响其他机器人的情况下,可以使系统快速地恢复可行性和最优性.但是该方法是一个不完备的方法,它不能保证对于所有可解问题都能找到解决方案.而在部分节点连通性未知的情况下,可建立方案树<sup>[47]</sup>,其中每个机器人的路径方案会随探测到的节点开闭情况产生不同的分支.然而,这种方法的计算复杂度会同时随机器人数量和连通性未知节点的数量呈指数型增长.

## 2.6 空间分解

碎片系统<sup>[48]</sup>将大规模的机器人工作空间划分为相对独立的小片区域,称之为碎片,碎片之间通过缓冲区连接.碎片系统的全局控制器接收用户提交的任务,通过设置各碎片的局部目标来分解全局任务.多个局部控制器并行地在碎片内部规划机器人路径,并通过缓冲区在碎片间转移机器人.每个碎片内部都可以根据机器人的实际状态和任务需求来更新其局部路径,而不影响其他碎片内的运行,因而易于处理机器人的动作失败或目标变化问题.碎片系统具备扩展性、鲁棒性和持续性,但是碎片系统不是完备的,在一些特殊的工作空间中,可能无法生成一个强连通的碎片图.碎片系统也不具备最优性,因为在一些复杂的地图中,无法保证碎片内部以及跨碎片

之间的路径质量.

鲁棒规划算法生成的路径方案可以在任务启动时一次性分配给各个机器人,成为它们的执行计划.然而,这些路径方案通常不能完全应对执行过程中遇到的不确定情况.特别是当意外事件超出了预定义的不确定性模型时,系统行为就会变得不可预测和不可控<sup>[19]</sup>.所以鲁棒规划方案通常需要在线执行策略进行实时协调,进一步保障任务的顺利执行.

## 3 面向不确定性的多机器人在线执行策略

面向不确定性的在线执行策略,是根据机器人的实际执行情况和实时环境信息,对因不确定性而可能失效的路径方案进行实时调整,以避免机器人之间产生冲突或死锁的重要手段.它不仅可以独立作用,也可以作为鲁棒规划方案在执行过程中的补充和完善,实现多个机器人在复杂动态环境中的协调移动.在线执行策略主要分为在线修复法、在线重规划法和其他方法3类.在线修复法通过命令机器人原地等待以使机器人之间恢复同步<sup>[28]</sup>或局部顺序<sup>[22,26]</sup>;在线重规划法旨在放弃失效的路径方案而重新寻找安全高效的路径<sup>[25,45]</sup>;其他方法主要通过特定的策略协调机器人的行进过程<sup>[56-57]</sup>.具体分类如表2所示.

### 3.1 在线修复法

移动延迟对于原有路径方案的破坏在于,它打乱了机器人之间协调好的时序关系,让本应该离开当前位置的机器人仍停留在原地,而与后续要进入当前位置的机器人之间发生冲突.为此,让未延迟的机器人

表2 面向不确定性的多机器人在线执行策略

类型	触发条件	调整方式	调整范围	场景特点	优点	缺点
在线修复法	任意延迟发生时	FSPs <sup>[28]</sup>	所有未延迟机器人等待	需要保持同步的场景,分布式	实现最简单,保持相对时序关系完全不变	单个机器人延迟就会造成全体等待,延迟频率高时整个系统将瘫痪
	因延迟导致机器人通过某一位置的局部顺序错乱时	TGP <sup>[21]</sup> , MCPs <sup>[28]</sup> , ADG <sup>[22]</sup> , RMTRACK <sup>[26-27]</sup>	部分机器人等待以恢复局部顺序	允许异步执行,分布式	只需部分机器人等待,较低的通信开销	部分机器人中仍可能出现不必要的等待,且难以应对故障机器人
	未来将会造成冲突的延迟发生时	SADG <sup>[58]</sup> , RMTRACK+TF <sup>[59]</sup> , MAPF-JSSP <sup>[60]</sup>	动态调整局部顺序	允许异步执行,集中式	进一步减少机器人不必要的等待,降低修复的时间代价	增大了计算负担,可能导致死锁以及后续冲突概率提高
在线重规划法	事件触发	鲁棒重规划 <sup>[32]</sup> , 边执行边规划 <sup>[22]</sup> , 优先级重规划 <sup>[62]</sup> , 重建式重规划 <sup>[63]</sup> , 先规划后优化 <sup>[64]</sup>	各方式不同	需对特定事件作出及时反应,集中式	规划及时,保证实时安全性,单次规划质量较高	单次规划的计算负担较大,规划次数不可控
	定期触发	滚动时域框架 <sup>[33,65]</sup> , 安全重规划 <sup>[29]</sup> , 带参数更新的重规划 <sup>[25]</sup>	各方式不同	环境动态性较强,集中式	单次规划的计算负担较小,能适应动态环境变化	两次规划间的空档期可能产生冲突,而延迟较少时又可能存在冗余规划
其他方法	机器人被阻碍时	Causal-PIBT <sup>[56]</sup> , 局部联合MDP <sup>[57]</sup>	受影响机器人	系统自由度较高,分布式	机器人通过本地交互实现分布式协调	协调效率较低

等待以修复时序关系的方法比较常用<sup>[66]</sup>,这种方法又可以分为修复同步性和修复局部顺序两类。

### 3.1.1 修复同步性

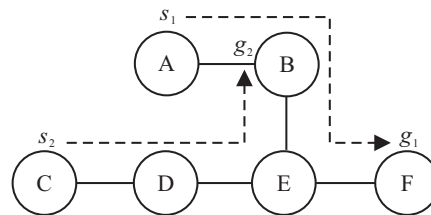
让所有当前时间步未延迟的机器人都在下一个时间步等待以保持同步,使执行过程中机器人的相对时序关系和规划方案保持完全一致。完全同步策略<sup>[28]</sup>(fully synchronized policies, FSPs)为机器人的状态变化过程分配统一的序号,机器人必须等待其他所有机器人都已经离开了各自的 $n$ 状态才能够进入 $n + 1$ 状态,这可以通过机器人在进入新状态时广播信息来实现。该方法虽然实现简单且保证同步,但只要有一个机器人延迟就会导致其他所有机器人等待,机器人数量较多或延迟频数较高时整个系统将会陷入瘫痪。

### 3.1.2 修复局部顺序

由于机器人间的冲突只会发生在路径的重叠部分,保证机器人有序通过重叠部分是避免冲突的一种方法。这种方法在多篇文献中以不同角度表述,但核心思想是类似的。文献<sup>[21]</sup>通过后处理将MRPP路径方案转化为有向无环的时序规划图(temporal plan graph, TGP)。TGP图中的点代表机器人进入一个位置点的事件,边代表事件间的依赖关系,只有满足了依赖关系才允许事件发生。TGP是保证机器人在重叠区域有序穿行的交通规则,不仅能够避免机器人在重叠区域发生冲突,而且能够避免因局部顺序被打乱后可能出现的死锁。基于TGP的最小通信策略<sup>[28]</sup>(minimal communication policies, MCPs)缩减了TGP中代表事件依赖的边的数量,减小了通信负担。进一步,文献<sup>[32]</sup>考虑了机器人之间无法进行实时通信的情况,即机器人每隔 $T$ 个时间步才通信它们所经历的延迟情况。为了保证在下一个通信间隔 $T$ 内不发生冲突,需要使机器人等待若干个时间步以保证修复后的路径方案仍是 $T$ -鲁棒的。

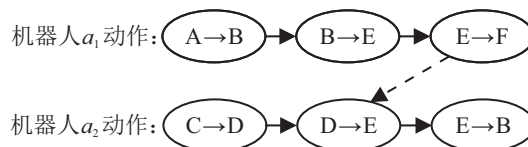
动作依赖图(action dependency graph, ADG)<sup>[22]</sup>是对TGP和MCPs的扩展,其特点在于将局部顺序建立在动作而非位置上。如图10所示,只有当机器人 $a_1$ 从E点到达F点之后,机器人 $a_2$ 才可以执行从D点到E点的动作。ADG允许非离散时间的一般化动作,而且只在机器人完成一个动作时进行通信,进一步减小了通信负担。但是若某个机器人出现长时间延迟,则可能会使依赖图中关联的许多未受延迟的机器人进入长时间等待,降低系统运行效率。为此文献<sup>[58]</sup>基于ADG提出了可以重排依赖关系的动作依赖图(switchable action dependency graph, SADG)。其核心

思想是基于机器人对延迟的实时反馈,在执行路径方案时动态调整机器人的动作依赖关系,并且使用混合整数线性规划求解依赖最优化问题。通过调整动作依赖,最大限度地减小延迟对整体任务完成时间的影响,从而提高执行效率和鲁棒性。但是对机器人局部顺序的重排可能会形成死锁<sup>[67]</sup>。类似地,基于作业车间调度的在线执行策略<sup>[60]</sup>也通过重新安排机器人到达某个节点的顺序来减小延迟对整体行进时间的影响。



(a) 任务地图

—→ 机器人自身的动作依赖  
 - - -→ 机器人之间的动作依赖



(b) 对应的动作依赖图

图10 动作依赖图<sup>[22,58]</sup>

同样的思想也出现在一些运动规划的文献中,例如鲁棒多机器人轨迹跟踪策略<sup>[26]</sup>(robust multi-robot trajectory tracking strategy, RMTRACK)将多机器人路径执行过程中的状态映射到协调空间中,并计算出协调空间中的冲突区域。通过对机器人的启停控制,保证实际执行过程中的状态线与原始规划的状态线是同伦的,即保持两条线位于协调空间中冲突区域的同侧。从MRPP的角度来看,就是保持机器人通过重叠区域的局部顺序与原方案一样。文献<sup>[68]</sup>进一步设计了使协调轨迹局部最优的RMTRACK导航器。与SADG之于ADG类似, RMTRACK+TF<sup>[59]</sup>也能够根据状态估计调换机器人通过重叠区域的局部顺序。另外,文献<sup>[69]</sup>研究了在线任务流场景下的优先级选择。研究证明如果任务是异步发布的,则静态优先级无法确保机器人都能到达目标位置;使用基于任务发布先后顺序的动态优先级可以避免发生死锁;使用基于到达重叠区域距离的启发式优先级可以减少不必要的等待时间。

机器人从产生等待需求开始,在其路径上可能有多个节点可选作等待时停留的位置。不同的选择

可能会对其他机器人的路径执行产生不同的影响,糟糕的选择可能会造成大面积的连锁等待.为此,文献[61]将机器人选择等待位置的问题(avoiding collisions by introducing delays, ACID)分离出来,使用CBS算法搜索最佳的等待位置,以使修复后的方案受到的总延迟最少.但ACID问题求解增大了计算负担,算法实时性较差.

在上述方法中,修复同步的优势在于保持了机器人之间的同步关系,除非有其他延迟发生,否则在下一个时间步时机器人间的相对时序关系与当前时间步完全一致,这在需要精确保持机器人相对时序关系的情况下是可取的.而修复局部顺序的好处在于可以大幅减少不必要的等待时间.总体而言,修复性策略只需要机器人具备最低程度的通信能力,并且可以平衡计算负担,因此具备较好的可扩展性.另外,修复性策略可以保留原方案的空间属性,这在一些对路径变化比较敏感的领域非常重要,如危险材料运输和空中交通管制等<sup>[61]</sup>.但是修复法会使机器人执行不必要的等待,特别是当延迟发生频数较高时,会严重损害方案执行的效率.

### 3.2 在线重规划法

在线重规划是指在执行过程中放弃剩余未执行的路径方案,以当前位置为起点,重新为一个或多个机器人规划路径的方法.相较于修复法,重规划法以较高的计算负担为代价,去除不必要的等待,获得更优的路径代价<sup>[62]</sup>.重规划法的另一个优势是能绕开无法通过的区域,包括故障机器人和动态障碍物等,提高了方案执行的可达性<sup>[64]</sup>.此外,重规划也是任务流异步发布场景下的主要手段<sup>[22,25]</sup>.如果执行过程中出现的延迟或故障较多,则重规划方法较为适用.重规划根据其触发条件,可以分为事件触发和定期触发的重规划.

#### 3.2.1 事件触发重规划

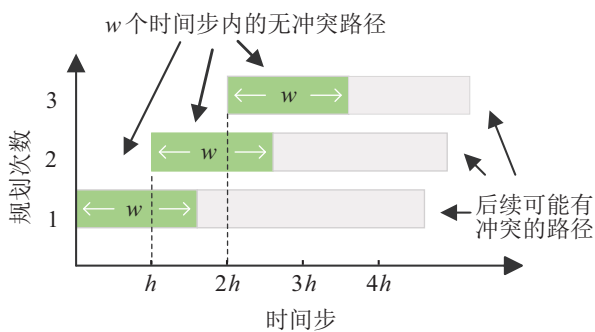
事件触发重规划是指只在特定事件发生时才进行重规划的方法.在针对 $k$ -鲁棒路径方案的鲁棒重规划方法<sup>[32]</sup>中,其触发事件被分为积极型、适度型和怠惰型3种,并针对性地给出重规划方案.其中怠惰型方法的逻辑在于未来的延迟可能消除当下延迟造成的预期冲突,实现较少的重规划次数.如果一个机器人检测到其路径上出现障碍物,则文献[22]的策略会让它停止移动,然后为它规划出一个新的方案.通过检测何时需要重规划,并在旧方案与新方案之间找到一个合适的切换点,可以在执行旧方案的同时并行地进行重规划<sup>[22]</sup>.这样既保证了在重规划期间机器

人不会停止移动,又保证了机器人动作的持续性.总体而言,事件触发重规划作用及时,能够保证机器人的实时安全,且单次规划质量较高,但也存在单次规划计算负担较大、规划次数不可控的问题.

除了触发事件的选择,重规划的策略也是影响重规划效率的重要因素.若直接采用离线规划算法对所有机器人进行重规划会带来巨大的计算负担,难以在实际工作环境中执行.对此,基于优先级的重规划方法<sup>[62]</sup>利用了优先级规划的解耦能力.当延迟发生时,首先找到与延迟机器人未来路径有重叠且局部顺序靠后的其他机器人,然后为这些机器人提高优先级并进行路径重规划,使它们能够优先穿过重叠区域或找到没有重叠的新路线.针对具体的重规划任务,文献[63]提出了一种先破坏再重建的策略,对于因不确定事件而无法执行的动作,先移除在它前后的一定数量的动作,再尝试在缺失的部分进行重规划,如果无法找到解,则扩大移除的比例.这种逐步扩大重规划比例的方法可以减小重规划的复杂度,并且使得路径方案的变动尽可能少.逐步扩大重规划的思想同样出现在文献[70]中,其设计了3个阶段的重规划.首先在第1阶段为被阻塞的机器人从阻塞点前的位置进行D\* lite寻路,利用D\* lite算法的动态性和记忆功能尽可能快地为单个机器人找到后续的可行路径.若第1阶段找到的路径中存在冲突,则在第2阶段将第1阶段路径方案作为CBS约束树的根节点进行冲突处理.若第2阶段仍未能找到无冲突路径,则在第3阶段清空现有路径方案,将所有机器人的当前状态作为起始状态,从零开始CBS规划.这种三阶段策略不仅提供了快速重规划的能力,还保证了重规划的可行性.另一种思路是先规划后优化<sup>[64]</sup>,当预先规划的路径被新出现的障碍物或其他机器人阻挡时,先利用离散图搜索算法找到一条绕过障碍物或其他机器人的路径,然后使用二次规划算法对路径进行优化,使其尽可能地接近原方案中的路径,并满足连续性、无碰撞和动力学限制等约束条件.由上可见,重规划策略应考虑规划范围和方法,并需要在规划能力与计算负担之间找到平衡.

#### 3.2.2 定期触发重规划

定期触发重规划是指每隔一段时间进行重规划.滚动时域框架<sup>[33,65]</sup>(receding horizon framework)要求每隔 $h$ 个时间步进行重规划,每次规划只解决 $w(w > h)$ 个时间步内的冲突,而忽略该时间范围之外的冲突,如图11所示.这种策略较好地平衡了单次重规划的计算负担和所需的安全性.

图 11 滚动时域框架<sup>[33,65]</sup>

安全重规划<sup>[29]</sup>假设机器人能够自行定位,并在到达每个位置点时与中心控制器进行通信,发送自己的时空信息.对于同时发送时空信息的一组机器人,中心控制器为它们基于实际状态进行重规划.这种做法可以消除保守的等待时间,使机器人能够更早到达目的地.该方法也可以在机器人只具备感知而不具备通信能力下执行,此时机器人可以缩减自身原计划的等待时间.文献[25]假设中心控制器知道每个机器人正在执行的命令,并且可以随时更新命令序列.新命令将在旧命令执行完毕后生效,并确保新命令的开始状态与旧命令的结束状态一致.新命令序列即新路径必须在旧指令执行完毕之前规划完成,否则机器人仍按旧命令执行,且新路径作废.文献[25]基于这种重规划规则在执行过程中定期更新所有机器人的路径,以减轻随机延迟的影响.并根据机器人执行过程中的实际延迟,使用贝叶斯推理更新预期延迟的概率分布,使算法能更好地适应环境的变化,提高路径规划的质量和鲁棒性.这种方法还适用于异步发布的任务流场景,当机器人在到达当前任务的目标点后,又分配新的任务并为其规划新的路径.

这类方法能够分散延迟对路径执行的影响,缩减规划时需要考虑的不确定性程度,同时能够根据实际执行情况,规划出更合理的路径方案.但是当延迟较多时,在两次重规划之间的空档期可能产生冲突,而当延迟较少时又可能存在冗余规划.

### 3.3 其他方法

此类方法主要依靠局部协调完成机器人在路径重叠部分的交通疏导<sup>[2,71]</sup>.不同于修复或重规划方法,文献[56]提出了一种时间无关的在线分布式控制策略 Causal-PIBT.该方法允许机器人异步行动,并将系统视为基于行动事件的过渡系统.事件包括单个机器人的收缩、请求和扩展,机器人在3种事件中交替,完成通向目标位置的协调运动.机器人在行动前被赋予唯一的优先级,如果高优先级个体被低优先级个体阻塞,则通过优先级继承和状态回溯机制,推动

低优先级个体为高优先级个体让路,以解决冲突和死锁问题. Causal-PIBT可以独立使用,也可以基于规划算法求得的初始方案,减轻由于在线策略的短视性带来的拥塞.实验表明,该方法相较于MCPs在延迟概率较高或规模较大的机器人系统中更具优势.文献[57]考虑了机器人在移动过程中偏离预期方向的问题,提出了使用局部联合MDP控制机器人驶离冲突区域的在线方法.此类方法灵活度较高,能够以局部协调的方式有效解决区域拥塞问题.

### 3.4 在线执行策略与鲁棒规划算法的结合

鲁棒规划算法为多机器人系统提供了对不确定性因素具有容忍能力的路径方案,而在线执行策略可以在路径方案的实际执行过程中实时协调机器人以应对超量干扰,两者的结合可以进一步提高多机器人系统的效率和安全性.例如:1)  $k$ -鲁棒路径规划算法和MCPs执行策略<sup>[32]</sup>.  $k$ -鲁棒的路径方案可以吸收  $k$  个时间步之内的延迟,而若延迟超过  $k$  个时间步,则依靠MCPs执行策略进行在线修复.2) UM\*算法和滚动时域框架<sup>[33]</sup>.其规划方案可以减小单个机器人发生冲突的概率,易于在滚动时域框架中对冲突概率变高的机器人进行重规划.3) CBS<sub>TU</sub>算法和定期重规划策略<sup>[29]</sup>.其规划方案为机器人设置了保守的时间约束,而定期重规划可以根据实际行进时间收紧时间约束,减少不必要的等待时间.4) Greedy STT-CBS算法和定期重规划策略<sup>[25]</sup>.其规划方案建立在延迟分布的初始预估上,而在执行过程中定期根据实际情况更新延迟分布.基于更新后的参数进行重规划,可以进一步提高安全性和效率.

## 4 面向不确定性的多机器人路径鲁棒规划的评价指标

针对不确定性下的多机器人路径鲁棒规划问题可以设置多种评估指标,其中安全指标主要是围绕冲突概率<sup>[25,31,33-34]</sup>建立的;而性能指标主要包括路径代价<sup>[28,32]</sup>和计算复杂度<sup>[19,34,56]</sup>等.这些指标不仅可以用于评估路径规划的安全性和性能,也引导了算法设计的方向和优化目标.

机器人的冲突概率是在不确定性模型中计算出的理论概率,主要包括局部冲突概率<sup>[25,34]</sup>和全局冲突概率<sup>[31,33]</sup>,计算方法主要包括数值积分<sup>[31,34]</sup>和蒙特卡洛<sup>[25,31,33-34]</sup>模拟.若两个机器人的路径在某一点或某条边上有重叠,则由于不确定的时间延迟,在这一点或边上会存在潜在冲突,这种潜在冲突的概率就是局部冲突概率.回顾第1节问题描述中的情况,令  $y = \Delta_i - \Delta_j$ ,则机器人  $a_i$  和  $a_j$  在点  $X$  的冲突概

率为

$$P(a_i, a_j, X) = P(A \cap B) = \int_y P(A|y)P(B|y)P(y)dy. \quad (11)$$

其中  $P(A|y)$  和  $P(B|y)$  可通过对服从 Gamma 分布的延迟时间概率密度函数求积分得到<sup>[34]</sup>.

由于在计算局部冲突概率的过程中,即使不确定延迟的发生是独立的,冲突也可能不是独立的<sup>[33]</sup>.也就是说,既需要考虑局部冲突概率也需要考虑全局冲突概率,这两种冲突概率的计算以数值积分法作为理论支撑,而实际的计算需要依靠蒙特卡洛模拟法(一种通过随机抽样试验求取统计值来推定未知量的常用方法).

性能指标中路径代价包括离线代价<sup>[31-32]</sup>和估计代价<sup>[28]</sup>,其中离线代价的计算方法与经典 MRPP 问题一致,如第1节中的式(1)和(2)所示,而估计代价是对不确定延迟下实际执行时间的估计.在服从几何分布的延迟模型<sup>[28]</sup>中,将机器人移动一步的时间设为  $1/(1-p_d)$ ,则机器人  $a_i$  进入某一状态  $x$  的时间等于状态  $x$  所有依赖状态  $x'$  中最大的预期时间加  $1/(1-p_d)$ ,即  $\tilde{\pi}_i(x) = \max_{j,x'}(\tilde{\pi}_j(x') + 1/(1-p_d))$ .由此,沿着时序规划图依次估算即可得到该方案的期望完工时间.而在服从 Gamma 分布的延迟模型<sup>[25,34]</sup>中,若不考虑机器人实际发生冲突后造成的额外延迟时间,则预估的行进时间总和等于各条路径上的基础行进时间和预估延迟时间的和.性能指标中的计算复杂度分析从不同角度展开,或与经典算法复杂度进行比较,例如 STT-CBS<sup>[34]</sup>、CBS<sub>TU</sub><sup>[29]</sup>以及单次重规划<sup>[70]</sup>的最差情况,其复杂度与 CBS 算法一致;或通过问题规约化来分析复杂度,例如 ACID<sup>[61]</sup>化为3色问题,OTIMAPP<sup>[19]</sup>化为3-SAT 问题;或做局部分析,例如对在线修复所依赖的 SADG<sup>[30]</sup>以及对 Causal-PIBT<sup>[56]</sup>中不同事件的复杂度进行分析;而大多数研究则是依靠仿真实验进行分析比较.

## 5 未来可发展的研究

面向不确定性的多机器人路径规划具有重要的理论研究价值,可以为多机器人系统提供更加智能化的规划和决策能力,使其在复杂任务和多样化场景下表现更为出色,为多机器人系统的广泛应用提供支撑.未来可从以下几个方面展开进一步研究:

1) 分布式规划与协调.随着机器人数量增加,集中式规划的难度呈指数级增长<sup>[9]</sup>,且执行过程中发生不确定事件的频率也相应增高,需要中心控制器进行协调的问题不断增多,给系统健康运行带来巨大负担

和挑战.分布式规划与协调算法可以部署到分布式系统中的各个机器人个体上,将问题解耦并分摊计算负担.一般而言,分布式系统中的个体只接入基本的地图信息和简单的初始路径,然后依靠自身有限的感知<sup>[72]</sup>与通信能力<sup>[73]</sup>进行在线独立决策.个体间可以根据设定的交通规则<sup>[56,73]</sup>或反应式控制<sup>[72]</sup>等方法进行协调,同时需要应对死锁和活锁等问题.此外,分布式方法也可以考虑划分工作区域以及优化任务分配,在任务层就减轻区域的交通拥堵,进一步提高系统扩展性和可靠性.

2) 结合强化学习的在线路径规划.当前使用的传统方法大多依赖固定的协调规则和启发式<sup>[1,60]</sup>,缺乏在复杂动态环境中的适应能力.而强化学习是基于环境交互的方法,通过学习环境的统计性质来做出决策,这一内在特性赋予了它对于环境变化的反应能力<sup>[74]</sup>.在强化学习中,MAPPP 问题通常被建模为部分可观察性的马尔科夫决策过程<sup>[75-76]</sup>,每个机器人依靠本地的观察结果及行动策略进行决策.通常,这样的行动策略由集中训练生成,训练过程中允许融合多个机器人的观察结果及专家引导<sup>[77-78]</sup>,再部署到本地机器人上,因此具备较好的扩展性.强化学习的应用范围包括无通信场景<sup>[79]</sup>及可通信场景<sup>[80]</sup>,其中优化通信机制<sup>[81]</sup>可以在通信消耗与协调需要间找到平衡.依靠全局规划器将任务分解并使用强化学习实现局部目标的混合策略<sup>[82]</sup>可以引导机器人走出稀疏奖励区,加快收敛速度.异步行动模型<sup>[83-84]</sup>能够应对执行过程中的时间不统一问题,但是其扩展性还需增强.此外,基于强化学习的方法更有利于扩展到任意角度和考虑运动学约束的场景中,但是需要进一步保证安全性和可达性<sup>[85]</sup>.

3) 异构多机器人系统的路径鲁棒规划.目前面向不确定性的多机器人路径规划主要考虑的是同构机器人系统,而异构多机器人是各类实际应用的发展需求.异构多机器人系统由多个类型不同但功能互补的机器人组成,可以拓展任务执行能力,如无人机与汽车的组合<sup>[86]</sup>、电子游戏中的异构单位小组<sup>[87]</sup>等.异构机器人具有不同的硬件尺寸,可以通过扩展机器人占用节点并增加节点约束的方式<sup>[88]</sup>进行处理.任务层方面,可以通过启发式方法实现异构任务分配与排序<sup>[89]</sup>,或通过分离异构任务将机器人按照不同功能进行分阶段规划<sup>[86]</sup>.为构建异构系统应对不确定性事件的鲁棒性,可以考虑与传统路径鲁棒规划算法结合,或采用强化学习方法<sup>[72,75]</sup>,然而,由于异构机器人系统的行为模式和不确定性更加复杂,算法

更加需要定制化的设计。

## 6 结论

本文从经典路径规划会遇到的不确定性问题出发,阐述了研究面向不确定性的多机器人路径鲁棒规划的重要性,并综述了相关解决办法。目前的研究主要从时间角度对多机器人路径规划中的不确定性进行建模,然后分别在规划阶段通过鲁棒规划算法,以牺牲路径性能为代价,换取对随机干扰的容忍性;在执行阶段通过在线执行策略实时协调机器人的移动,避免意外冲突。本文对上述算法和策略进行详细梳理,给出不同切入点下的方法分类和整理,分析其逻辑与特点并总结相关评价指标。最后,对多机器人路径鲁棒规划研究关键技术进行展望,即未来可从分布式规划与协调、结合机器学习的在线路径规划、异构多机器人系统的鲁棒路径规划等方面开展研究。

### 参考文献(References)

- [1] Stern R, Sturtevant N, Felner A, et al. Multi-agent pathfinding: Definitions, variants, and benchmarks[J]. Proceedings of the International Symposium on Combinatorial Search, 2021, 10(1): 151-158.
- [2] Street C. Towards multi-robot coordination under temporal uncertainty[C]. Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems. Auckland, 2020: 2217-2218.
- [3] Varambally S, Li J Y, Koenig S. Which MAPF model works best for automated warehousing?[J]. Proceedings of the International Symposium on Combinatorial Search, 2022, 15(1): 190-198.
- [4] Laurent F, Schneider M, Scheller C, et al. Flatland competition 2020: MAPF and MARL for efficient train coordination on a grid world[C]. NeurIPS 2020 Competition and Demonstration Track. Virtual: ML Research Press, 2021, 133: 275-301.
- [5] Morris R, Păsăreanu C, Luckow K S, et al. Planning, scheduling and monitoring for airport surface operations[C]. AAAI Workshop: Planning for Hybrid Systems. Menlo Park: AAAI, 2016: 608-614.
- [6] Stern R. Multi-agent path finding-an overview[C]. Artificial Intelligence: 5th RAAI Summer School. Dolgoprudny, 2019: 96-115.
- [7] Wurman P R, Andrea R, Mountz M. Coordinating hundreds of cooperative, autonomous vehicles in warehouses[J]. AI Magazine, 2008, 29(1): 9-20.
- [8] Ma H. Graph-based multi-robot path finding and planning[J]. Current Robotics Reports, 2022, 3(3): 77-84.
- [9] Yu J J, LaValle S M. Structure and intractability of optimal multi-robot path planning on graphs[C]. Proceedings of the 27th AAAI Conference on Artificial Intelligence. Bellevue, 2013: 1443-1449.
- [10] Silver D. Cooperative pathfinding[C]. Proceedings of the First AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment. Marina del Rey, 2005: 117-122.
- [11] Phillips M, Likhachev M. SIPP: Safe interval path planning for dynamic environments[C]. IEEE International Conference on Robotics and Automation. Shanghai, 2011: 5628-5635.
- [12] Sharon G, Stern R, Felner A, et al. Conflict-based search for optimal multi-agent path finding[J]. Artificial Intelligence, 2015, 219: 40-66.
- [13] Okumura K, Machida M, Défago X, et al. Priority inheritance with backtracking for iterative multi-agent path finding[J]. Artificial Intelligence, 2022, 310: 103752.
- [14] 张凯翔, 毛剑琳, 向凤红, 等. 基于讨价还价博弈机制的B-IHCA\*多机器人路径规划算法[J]. 自动化学报, 2023, 49(7): 1483-1497.  
(Zhang K X, Mao J L, Xiang F H, et al. B-IHCA\*, a bargaining game based multi-agent path finding algorithm[J]. Acta Automatica Sinica, 2023, 49(7): 1483-1497.)
- [15] 张丹露, 孙小勇, 傅顺, 等. 智能仓库中的多机器人协同路径规划方法[J]. 计算机集成制造系统, 2018, 24(2): 410-418.  
(Zhang D L, Sun X Y, Fu S, et al. Cooperative path planning in multi-robots for intelligent warehouse[J]. Computer Integrated Manufacturing Systems, 2018, 24(2): 410-418.)
- [16] 宣志玮, 毛剑琳, 张凯翔. CBS框架下面向复杂地图的低拓展度A\*算法[J]. 电子学报, 2022, 50(8): 1943-1950.  
(Xuan Z W, Mao J L, Zhang K X. Low-expansion A\* algorithm based on CBS framework for complex map[J]. Acta Electronica Sinica, 2022, 50(8): 1943-1950.)
- [17] 张洪琳, 吴耀华, 胡金昌, 等. 一种基于改进冲突搜索的多机器人路径规划算法[J]. 控制与决策, 2023, 38(5): 1327-1335.  
(Zhang H L, Wu Y H, Hu J C, et al. A multi-robot path finding algorithm based on improved conflict search[J]. Control and Decision, 2023, 38(5): 1327-1335.)
- [18] Švancara J, Vlk M, Stern R, et al. Online multi-agent pathfinding[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2019, 33(1): 7732-7739.
- [19] Okumura K, Bonnet F, Tamura Y, et al. Offline time-independent multiagent path planning[J]. IEEE Transactions on Robotics, 2023, 39(4): 2720-2737.
- [20] Ulusoy A, Smith S, Ding X, et al. Robust multi-robot optimal path planning with temporal logic constraints[C]. Proceedings of the 2012 IEEE International Conference

- on Robotics and Automation. Piscataway: IEEE, 2012: 4693-4698.
- [21] Hönig W, Kumar T, Cohen L, et al. Multi-agent path finding with kinematic constraints[C]. Proceedings of the International Conference on Automated Planning and Scheduling. Menlo Park: AAAI, 2016, 26: 477-485.
- [22] Honig W, Kiesel S, Tinka A, et al. Persistent and robust execution of MAPF schedules in warehouses[J]. IEEE Robotics and Automation Letters, 2019, 4(2): 1125-1131.
- [23] Tomy M, Lacerda B, Hawes N, et al. Battery charge scheduling in long-life autonomous mobile robots via multi-objective decision making under uncertainty[J]. Robotics and Autonomous Systems, 2020, 133: 103629.
- [24] Okumura K, Tixeuil S. Fault-tolerant offline multi-agent path planning[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2023, 37(10): 11647-11654.
- [25] Kita A, Suenari N, Okada M, et al. Online re-planning and adaptive parameter update for multi-agent path finding with stochastic travel times[C]. Proceedings of the 2023 International Conference on Autonomous Agents and Multi-agent Systems. New York: ACM, 2023: 2556-2558.
- [26] Čáp M, Gregoire J, Frazzoli E. Provably safe and deadlock-free execution of multi-robot plans under delaying disturbances[C]. IEEE/RSJ International Conference on Intelligent Robots and Systems. New York: ACM, 2016: 5113-5118.
- [27] Qin W, Gomez M, Orosz G. Stability and frequency response under stochastic communication delays with applications to connected cruise control design[J]. IEEE Transactions on Intelligent Transportation Systems, 2016, 18(2): 388-403.
- [28] Ma H, Kumar T, Koenig S. Multi-agent path finding with delay probabilities[C]. Proceedings of the AAAI Conference on Artificial Intelligence. Menlo Park: AAAI, 2017, 31(1): 3605-3612.
- [29] Shahar T, Shekhar S, Atzmon D, et al. Safemulti-agent path finding with time uncertainty[J]. Journal of Artificial Intelligence Research, 2021, 70: 923-954.
- [30] Naderpour M, Lu J, Zhang G Q. An intelligent situation awareness support system for safety-critical environments[J]. Decision Support Systems, 2014, 59: 325-340.
- [31] Atzmon D, Stern R, Felner A, et al. Probabilistic robust multi-agent path finding[J]. Proceedings of the International Conference on Automated Planning and Scheduling, 2020, 30: 29-37.
- [32] Atzmon D, Stern R, Felner A, et al. Robust multi-agent path finding and executing[J]. Journal of Artificial Intelligence Research, 2020, 67: 549-579.
- [33] Wagner G, Choset H. Path planning for multiple agents under uncertainty[J]. Proceedings of the International Conference on Automated Planning and Scheduling, 2017, 27: 577-585.
- [34] Peltzer O, Brown K, Schwager M, et al. STT-CBS: A conflict-based search algorithm for multi-agent path finding with stochastic travel times[J/OL]. 2021, arXiv: 2004.08025.
- [35] Fu B, Smith W, Rizzo D M, et al. Robust task scheduling for heterogeneous robot teams under capability uncertainty[J]. IEEE Transactions on Robotics, 2023, 39(2): 1087-1105.
- [36] Chen Z, Harabor D, Li J, et al. Symmetry breaking for  $k$ -robust multi-agent path finding[C]. Proceedings of the AAAI Conference on Artificial Intelligence. Menlo Park: AAAI, 2021, 35(14): 12267-12274.
- [37] 李昊楠, 毛剑琳, 张凯翔, 等. 一种基于安全区间的多机器人路径  $k$  鲁棒规划算法[J]. 仪器仪表学报, 2023, 44(10): 274-282.  
(Li H N, Mao J L, Zhang K X, et al. Multi-robot path  $k$  robust planning algorithm based on safe interval[J]. Chinese Journal of Scientific Instrument, 2023, 44(10): 274-282.)
- [38] Zhang K X, Mao J L, Zhang S F, et al. A priority-based hierarchical framework for  $k$ -robust multi-agent path finding[J]. IEEE Transactions on Intelligent Vehicles, 2024: 1-11.
- [39] Lodigiani G. Robustness in multi-agent pickup and delivery with delays[D]. Milan: Politecnico di Milano, 2021.
- [40] Li J, Zhang H, Gong M, et al. Departure scheduling and taxiway path planning under uncertainty[C]. AIAA Aviation 2019 Forum. Dallas: AIAA, 2019: 2930-2936.
- [41] Omidshafiei S, Agha-Mohammadi A A, Amato C, et al. Decentralized control of multi-robot partially observable Markov decision processes using belief space macro-actions[J]. The International Journal of Robotics Research, 2017, 36(2): 231-258.
- [42] Floriano B, Borges G A, Ferreira H. Planning for decentralized formation flight of UAV fleets in uncertain environments with dec-POMDP[C]. International Conference on Unmanned Aircraft Systems. Atlanta, 2019: 563-568.
- [43] Oliehoek F, Witwicki S, Kaelbling L. A sufficient statistic for influence in structured multiagent environments[J]. Journal of Artificial Intelligence Research, 2021, 70: 789-870.
- [44] Mansouri M, Lacerda B, Hawes N, et al. Multi-robot planning under uncertain travel times and safety constraints[C]. Proceedings of the 28th International Joint Conference on Artificial Intelligence. New York: ACM, 2019: 478-484.

- [45] Street C, Lacerda B, Mühlig M, et al. Multi-robot planning under uncertainty with congestion-aware models[C]. Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems. Auckland, 2020: 1314-1322.
- [46] Nekvinda M, Bartak R. Contingent planning for robust multi-agent path finding[C]. IEEE 33rd International Conference on Tools with Artificial Intelligence. Washington, 2021: 487-492.
- [47] Shofer B, Shani G, Stern R. Multi agent path finding under obstacle uncertainty[C]. Proceedings of the 33rd International Conference on Automated Planning and Scheduling. New York: ACM, 2023: 402-410.
- [48] Leet C, Li J, Koenig S. Shardsystems: Scalable, robust and persistent multi-agent path finding with performance guarantees[C]. Proceedings of the AAAI Conference on Artificial Intelligence. Menlo Park: AAAI, 2022: 9386-9395.
- [49] Atzmon D, Diei A, Rave D. Multi-train path finding[J]. Proceedings of the International Symposium on Combinatorial Search, 2021, 10(1): 125-129.
- [50] Chen Z, Li J Y, Harabor D, et al. Multi-train path finding revisited[J]. Proceedings of the International Symposium on Combinatorial Search, 2022, 15(1): 38-46.
- [51] Tihanyi D, Lu Y M, Karaca O, et al. Multi-robot task allocation for safe planning against stochastic hazard dynamics[C]. European Control Conference. Bucharest, 2023: 1-6.
- [52] Street C, Lacerda B, Staniaszek M, et al. Context-aware modelling for multi-robot systems under uncertainty[C]. Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems. New York: ACM, 2022: 1228-1236.
- [53] Street C, Mansouri M, Lacerda B. Formal modelling for multi-robot systems under uncertainty[J]. Current Robotics Reports, 2023, 4(3): 55-64.
- [54] Wang X, Liu J, Nugent C, et al. Formal verification for multi-agent path execution under stochastic environments[EB/OL]. (2023-07-29)[2023-11-08]. <https://ssrn.com/abstract=4524943>.
- [55] Buchholz P, Kriege J, Felko I. Input modeling with phase-type distributions and Markov models: Theory and applications[M]. Cham: Springer International Publishing, 2014.
- [56] Okumura K, Tamura Y, Défago X. Time-independent planning for multiple moving agents[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2021, 35(13): 11299-11307.
- [57] Levy E, Shani G, Stern R. An online approach for multi-agent path finding under movement uncertainty (extended abstract)[J]. Proceedings of the International Symposium on Combinatorial Search, 2022, 15(1): 299-301.
- [58] Berndt A, Duijkeren N V, Palmieri L, et al. A feedback scheme to reorder a multi-agent execution schedule by persistently optimizing a switchable action dependency graph[C]. Proceedings of the 6th Workshop on Distributed and Multi-Agent Planning. Menlo Park: AAAI, 2020, 6: 1-9.
- [59] Adem C, Jason M. Online plan repair in multi-robot coordination with disturbances[C]. Proceedings of the 2019 IEEE International Conference on Robotics and Automation. Piscataway: IEEE, 2019: 3333-3339.
- [60] Zahrádka D, Kubišta D, Kulich M. Solving robust execution of multi-agent pathfinding plans as a scheduling problem[C]. ICAPS'23 Workshop: Planning and Robotics. Menlo Park: AAAI, 2023: 1-11.
- [61] Köttinger J, Almagor S, Salzman O, et al. Introducing delays in multi-agent path finding[J/OL]. 2023, arXiv: 2307.11252.
- [62] Li J, Zhe C, Zheng Y, et al. Scalablerrail planning and replanning: Winning the 2020 flatland challenge[C]. Proceedings of the International Conference on Automated Planning and Scheduling. Menlo Park: AAAI, 2021, 31: 477-485.
- [63] Bechon P, Lesire C, Barbier M. Hybrid planning and distributed iterative repair for multi-robot missions with communication losses[J]. Autonomous Robots, 2020, 44(3): 505-531.
- [64] Senbaslar B, Hönig W, Ayanian N. Robust trajectory execution for multi-robot teams using distributed real-time replanning[C]. Distributed Autonomous Robotic Systems. Cham: Springer, 2019: 167-181.
- [65] Li J, Tinka A, Kiesel S, et al. Lifelong multi-agent path finding in large-scale ware houses[C]. Proceedings of the AAAI Conference on Artificial Intelligence. Menlo Park: AAAI, 2023, 35(13): 11272-11281.
- [66] Felner A, Stern R, Rosenschein J S, et al. Searching for close alternative plans[J]. Autonomous Agents and Multi-Agent Systems, 2007, 14(3): 211-237.
- [67] Coskun A. Multi-robot coordination with environmental disturbances[D]. Columbia: University of South Carolina, 2021.
- [68] Gregoire J, Čáp M, Frazzoli E. Locally-optimal multi-robot navigation under delaying disturbances using homotopy constraints[J]. Autonomous Robots, 2018, 42(4): 895-907.
- [69] Mannucci A, Pallottino L, Pecora F. On provably safe and live multi robot coordination with online goal posting[J]. IEEE Transactions on Robotics, 2021, 37(6): 1973-1991.
- [70] Semiz F, Polat F. Incremental multi-agent path finding[J]. Future Generation Computer Systems, 2021, 116: 220-233.

- [71] Brown K. Algorithms for multi-agent robotic assembly planning[D]. Stanford: Stanford University, 2021.
- [72] Zhu H, Brito B, Alonso-Mora J. Decentralized probabilistic multi-robot collision avoidance using buffered uncertainty-aware Voronoi cells[J]. *Autonomous Robots*, 2022, 46(2): 401-420.
- [73] Maoudj A, Christensen A L. Improved decentralized cooperative multi-agent path finding for robots with limited communication[J]. *Swarm Intelligence*, 2023: 1-19.
- [74] Wong A, Bäck T, Kononova A V, et al. Deep multiagent reinforcement learning: Challenges and directions[J]. *Artificial Intelligence Review*, 2023, 56(6): 5023-5056.
- [75] Amato C, Konidaris G, Kaelbling L P, et al. Modeling and planning with macro-actions in decentralized POMDPs[J]. *Journal of Artificial Intelligence Research*, 2019, 64: 817-859.
- [76] Floriano B R O, Borges G A, Ferreira H C, et al. Hybrid dec-POMDP/PID guidance system for formation flight of multiple UAVs[J]. *Journal of Intelligent & Robotic Systems*, 2021, 101(3): 65.
- [77] Sartoretti G, Kerr J, Shi Y F, et al. PRIMAL: Pathfinding via reinforcement and imitation multi-agent learning[J]. *IEEE Robotics and Automation Letters*, 2019, 4(3): 2378-2385.
- [78] Damani M, Luo Z Y, Wenzel E, et al. PRIMAL<sub>2</sub>: Pathfinding via reinforcement and imitation multi-agent learning-lifelong[J]. *IEEE Robotics and Automation Letters*, 2021, 6(2): 2666-2673.
- [79] Choi J, Lee G, Lee C. Reinforcement learning-based dynamic obstacle avoidance and integration of path planning[J]. *Intelligent Service Robotics*, 2021, 14(5): 663-677.
- [80] Zhu C X, Dastani M, Wang S H. A survey of multi-agent deep reinforcement learning with communication[J]. *Autonomous Agents and Multi-Agent Systems*, 2024, 38(1): 4.
- [81] Ma Z Y, Luo Y D, Pan J. Learning selective communication for multi-agent path finding[J]. *IEEE Robotics and Automation Letters*, 2022, 7(2): 1455-1462.
- [82] Liu Z X, Chen B M, Zhou H Y, et al. MAPPER: Multi-agent path planning with evolutionary reinforcement learning in mixed dynamic environments[C]. 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems. Las Vegas, 2020: 11748-11754.
- [83] Xiao Y, Tan W, Amato C. Asynchronous actor-critic for multi-agent reinforcement learning[J]. *Advances in Neural Information Processing Systems*, 2022, 35: 4385-4400.
- [84] Yu C, Yang X Y, Gao J X, et al. Asynchronous multi-agent reinforcement learning for efficient real-time multi-robot cooperative exploration[C]. *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*. London, 2023: 1107-1115.
- [85] Semnani S H, Liu H, Everett M, et al. Multi-agent motion planning for dense and dynamic environments via deep reinforcement learning[J]. *IEEE Robotics and Automation Letters*, 2020, 5(2): 3221-3226.
- [86] Choudhury S, Solovey K, Kochenderfer M, et al. Coordinated multi-agent pathfinding for drones and trucks over road networks[C]. *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*. New York: ACM, 2022: 272-280.
- [87] Ma H, Yang J X, Cohen L, et al. Feasibility study: Moving non-homogeneous teams in congested video game environments[J]. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 2021, 13(1): 270-272.
- [88] Li J Y, Surynek P, Felner A, et al. Multi-agent path finding for large agents[J]. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019, 33(1): 7627-7634.
- [89] Sadeghi A, Smith S L. Heterogeneous task allocation and sequencing via decentralized large neighborhood search[J]. *Unmanned Systems*, 2017, 5(2): 79-95.

## 作者简介

张书凡(1994—),男,博士生,主要研究方向为多机器人路径规划, E-mail: 20233103006@stu.kust.edu.cn;

毛剑琳(1976—),女,教授,博士,博士生导师,主要研究方向为通信网络资源分配与协议优化、多机器人系统协同控制, E-mail: jlmao@kust.edu.cn;

张凯翔(1993—),男,博士生,主要研究方向为多机器人调度和路径规划, E-mail: kaixiangzhang@stu.kust.edu.cn;

李睿祺(1996—),男,博士生,主要研究方向为多机器人路径规划, E-mail: 1106634086@qq.com;

李大炎(1981—),男,实验师,硕士,主要研究方向为嵌入式系统设计与应用, E-mail: lidayan@kust.edu.cn;

王妮娅(1978—),女,实验师,硕士,主要研究方向为电路与系统、电气控制, E-mail: 23208170@qq.com.